Licenciatura em Eng.ª Informática

Computação Distribuida - 3º Ano - 1º Semestre

Trabalho Nº 2: Java RMI

1. Introdução

O objectivo deste trabalho consiste em adicionar vários servidores de Front-End ao servidor inicial utilizando o Java RMI. Esta nova versão do trabalho deve assegurar a integração com o trabalho anterior (Nº1). Este trabalho deverá ser feito individualmente ou em grupos de dois ou três alunos.

2. Descrição do trabalho:

Este trabalho insere vários Front-Ends que interceptam a comunicação entre o servidor principal e os utilizadores/votantes, como representado na figura 1:
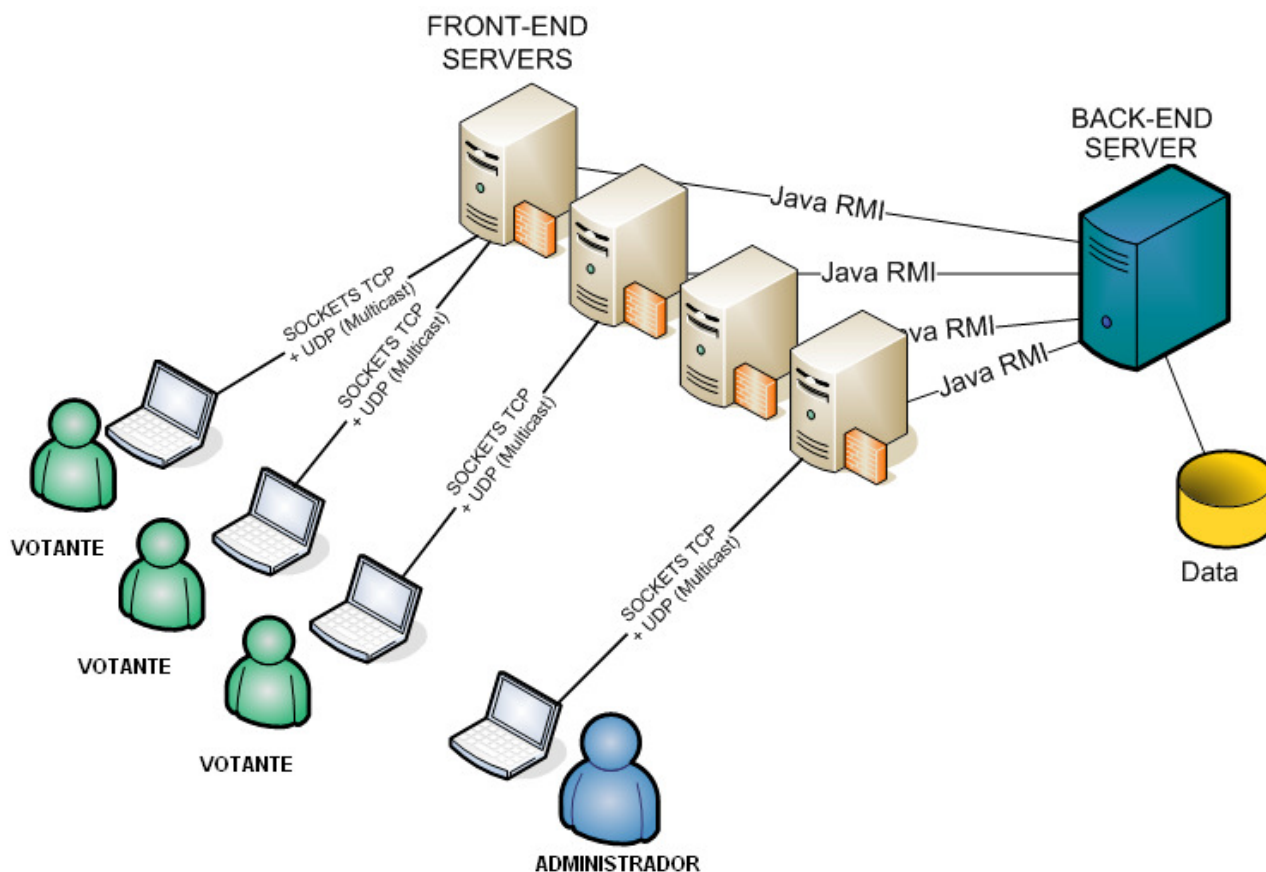
Figura 1: Arquitectura da aplicação, utilizando Java RMI

O propósito dos servidores Front-End é o de aliviar a carga a que pode estar sujeito o servidor principal (Back-End). Neste tipo de arquitectura, os Front-Ends são utilizados para disponibilizar conteúdo estático aos utilizadores, tais com páginas web, enquanto o servidor principal trata dos pedidos necessários para a componente dinâmica das páginas web. Isto significa que os Front-Ends podem ter em cache alguma informação estática do

servidor principal, podendo disponibilizar esta informação directamente aos clientes, sem a necessidade de consultar o servidor principal para responder aos pedidos. A aplicação dos utilizadores e administradores devem ter um ficheiro de configuração com os endereços dos servidores Front-End conhecidos. Quando a aplicação inicia, ela deve escolher os servidores Front-End pela ordem na lista.

3. Falhas na conectividade com o servidor

A sua aplicação deve tolerar falhas na comunicação do canal TCP. Você já terminou a parte da manipulação de excepções no primeiro trabalho. Agora, você precisa de estender esta capacidade de modo a que o cliente consiga mudar o servidor de Front-End activo caso este esteja indisponível: se o cliente recebe uma excepção na comunicação com o servidor Front-End, deve haver novas tentativas (por exemplo, 5 tentativas em 5 segundos) para conectar novamente com o servidor. Se depois das tentativas o servidor ainda não estiver disponível, o cliente tentará a conexão com um outro servidor Front-End da lista. O cliente deve repetir este processo até alcançar um servidor Front-End activo ou até que não haja mais servidores a pesquisar na lista.

O servidor principal (Back-End Server) pode igualmente falhar e, neste caso, os servidores Front-End devem assegurar as excepções do RMI e devem tentar nova conexão periodicamente. Neste caso os servidores Front-End nunca desistem de conectar e esperam o restabelecimento do servidor RMI.

Para finalidades de teste, os servidores Front-End e Back-End devem incluir uma consola para receber os seguintes comandos: *halt* (parar de responder), *restart* (reiniciar) e *exit* (sair). Todos os comandos devem assegurar que os servidores (em especial o servidor Back-End) permaneçem num estado consistente. O mecanismo de fail-over deve ser exaustivamente testado para garantir que a aplicação se comporta correctamente sob diversas falhas nos servidores. Nenhuma mensagem dos clientes deve perder-se durante o processo de re-conectar e as falhas dos servidores Front-End devem ser totalmente transparentes para o utilizador final, que continuará a interagir com o serviço de votação.

4. Modificações às aplicações do trabalho Nº 1

Neste trabalho, as aplicações necessitam de um tratamento novo das excepções como explicado anteriormente. O resto das aplicações permanece na maior parte inalterado. Isto significa que ainda utilizam o TCP e UDP tal como faziam anteriormente.

Por outro lado, o servidor Front-End necessita de algumas modificações para suportar a comunicação RMI com os servidores Back-End.

5. Interface gráfico

Novamente, os alunos não devem perder tempo no desenvolvimento de interfaces gráficos.

6. Fases seguintes

No trabalho Nº 3, os alunos devem desenvolver uma solução de conectividade directa ao servidor Back-End utilizando um browser e JSP/JavaBeans. No trabalho Nº 4, o objectivo

será a utilização do SOAP na comunicação entre o servidor Back-End e aplicações de Back-Office .

7. Relatório do Trabalho

Além do software desenvolvido deverá ainda escrever um relatório do trabalho prático. O relatório deverá ter as seguintes secções:

- Tempo médio (horas) utilizado por cada elemento do grupo no trabalho
- Introdução
- Arquitectura interna dos Front-Ends
- Tratamento de falhas transitórias nos sockets e RMI
- Manual de utilização
- Manual de instalação e configuração
- Descrição dos testes efectuados à aplicação

8. Prazo de Entrega do Trabalho

O prazo para a entrega dos trabalhos é de quatro semanas a contar da aula em que o enunciado foi apresentado. Não serão aceites trabalhos fora do prazo. A entrega deverá ser feita por e-mail num ficheiro zip (ou rar) contendo o relatório (**obrigatório**) em PDF e as listagens dos programas realizados, obedecendo OBRIGATORIAMENTE ao seguinte formato:

`a123456-trabalho-2-CD.zip`

9. Referências

SUN - Essentials of the Java Programming Language
http://java.sun.com/developer/onlineTraining/Programming/BasicJava1
http://java.sun.com/developer/onlineTraining/Programming/BasicJava2

SUN - The Java Tutorials, Networking
http://java.sun.com/docs/books/tutorial/networking/index.html

SUN - Remote Method Invocation Home
http://java.sun.com/javase/technologies/core/basic/rmi/